LISTING OF THE CLAIMS

1.   (Currently Amended) A method comprising:

establishing communication between a plurality of non-Java-based server nodes ~~of a first instance~~ and a plurality of Java-based server nodes ~~of a second instance~~ via an intermediate server;

generating a packet, on ~~one of the~~ a first non-Java-based server nodes, to be transmitted from ~~one of the~~ first non-Java-based server nodes to ~~one of the~~ a first Java-based server nodes;

specifying in a header of the packet an address of ~~a destination~~ the first Java-based server node and information that indicates that the packet is generated by ~~one of the~~ the first non-Java-based server nodes;

forwarding the packet to the intermediate server from the ~~one of the~~ first non-Java-based server nodes;

forwarding the packet to the ~~destination~~ first Java-based server node from the intermediate server based on the address provided in the header of the packet, wherein ~~by synchronizing~~ the packet header is formatted to be compatible with the format of the first Java-based server node such that it can be decoded by the first Java-based server node ~~a destination server~~;

maintaining a list of services, the list of services includes processes and tasks performed by the Java-based server nodes; and

maintaining a list of services, the list of services includes processes and tasks performed by the non-Java-based server nodes, wherein the maintained lists of services facilitate communications between the first ~~-~~Java-based server nodes and first Java-based server nodes.

2.   (Currently Amended) The method of claim 1, further comprising:

generating a second packet to be transmitted from ~~one of the~~ a second Java-based server nodes to ~~one of the~~ a second non-Java-based server nodes;

specifying in a header of the second packet an address of ~~a destination~~ the second non-Java-based server node and information that indicates that the packet is generated by ~~one of~~ the second Java-based server nodes;

forwarding the second packet to the intermediate server from the ~~one of the~~ second Java-based server nodes; and

forwarding the second packet to the ~~destination~~ second non-Java-based server node from the intermediate server based on the address provided in the header of the second packet.

3.   (Currently Amended) The method of claim 1~~2~~, further comprising:

sending notification of a status of each of the listed services to the non-Java-based server nodes ~~in the first instance~~ such that the maintained lists of services can be updated.

4.   (Currently Amended) The method of claim 1~~3~~, further comprising:

sending notification of a status of each of the listed services to the Java-based server nodes ~~in the second instance~~ such that the maintained lists of services can be updated.

5.   (Currently Amended) The method of claim 1~~4~~, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server, the intermediate server providing interoperability of communications between the Java based server nodes and the non-Java based server nodes.

6.   (Original) The method of claim 1, further comprising:

implementing Java 2 Platform Enterprise Edition (J2EE) applications in the Java-based server nodes.

7.   (Currently Amended) A system comprising:

~~a first instance including~~ a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from ~~one of the~~ a first non-Java-based server node~~s~~;

~~a second instance including~~ a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from ~~one of the~~ a first Java-based server node~~s~~; ~~and~~

a message server coupled between the ~~first~~ Java-based server nodes and ~~second instances~~ the non-Java-based server nodes to establish communication ~~there~~ between the first Java-based server node and the first non-Java-based server node ~~the first instance and the second instance~~ by ~~synchronizing~~ formatting each packet header such that it ~~can be decoded by~~ is compatible with a format of a destination server; and

an enqueue server coupled between the Java-based server nodes and the non-Java based server nodes to provide central locking services to lock access to resources in the system for use during communications between the first Java-based server node and the first non-Java based server node such that communications between the first Java-based server node and the first non-Java based server node are not interrupted.

8. (Currently Amended) The system of claim 7, ~~wherein each of the instances~~ further comprising ~~es~~ a dispatcher to distribute client requests for services to the Java-based server nodes and the non-Java based server nodes ~~server nodes of the respective instance~~, such that the dispatcher distributes the load amongst the Java-based server nodes and the non-Java based server nodes during communications.

9. (Currently Amended) The system of claim 7, wherein the message server provides communications between the Java-based server nodes and the non-Java based server nodes ~~is to~~ by rout~~ing~~ing message packets between the non-Java-based server nodes ~~of the first instance~~ and the Java-based server nodes ~~of the second instance~~.

10. (Currently Amended) The system of claim 7, wherein the message server is to assign a service identification associated with each type of services executed on the server nodes, wherein assigning the service identifications assists in providing communications between the Java-based server nodes and the non-Java based server nodes.

11. (Currently Amended) The system of claim 10, wherein the message server includes a service repository to maintain a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in

providing communications between the Java-based server nodes and the non-Java based server nodes.

12.   (Currently Amended) The system of claim ~~7~~8, wherein the message server further comprises:

a first repository to maintain a list of services currently being executed on the non-Java-based ~~instances~~server nodes; and

a second repository to maintain a list of services currently being executed on the Java-based ~~instances~~server nodes, wherein the first and second repositories facilitate distribution of load by indicating currently executing services.

13.   (Currently Amended) The system of claim ~~7~~8, wherein the message server is to maintain a list of services performed by the Java-based server nodes and the non-Java based server nodes~~the instances~~ and a status corresponding to each of the listed services, and to send notification of the status of the listed services to ~~the instances~~the dispatcher such that the dispatcher distributes the load in each instance following requests for services.

14.   (Original) The system of claim 7, wherein the Java-based instances are capable of implementing Java 2 Platform Enterprise Edition (J2EE) applications.

15.   (Currently Amended) A message server comprising:

a first communication interface to establish communication with a plurality of non-Java-based server nodes;

a second communication interface to establish communication with a plurality of Java-based server nodes; and

a controller to transfer packets between the non-Java-based server nodes and the Java-based server nodes, the controller to ensure the packets are received by a destination server node by resending the packets if a confirmation of receipt has not been received from the destination server node.

16.   (Currently Amended) The message server of claim 15, wherein the controller is to assign a service identification associated with each type of services executed on the <u>Java based</u> server nodes <u>and non-Java-based server nodes, wherein assigning the service identifications assists in providing communications between the first and second communications interfaces</u>.

17.   (Currently Amended) The message server of claim 16, further comprising:
       a service repository maintain a list of the assigned service identification and corresponding service names<u>, wherein maintaining the service identifications and the service names assists in providing communications between the first and second communications interfaces</u>.

18.   (Currently Amended) The message server of claim 15, further comprising:
       a first repository to maintain a list of services currently being executed on the non-Java-based server nodes; and
       a second repository to maintain a list of services currently being executed on the Java-based server nodes<u>, wherein the first and second repositories facilitate distribution of load by indicating currently executing services</u>.

19.   (Currently Amended) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the first repository to the non-Java-based server nodes <u>such that the maintained lists of services can be updated</u>.

20.   (Currently Amended) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the second repository to the Java-based server nodes <u>such that the maintained lists of services can be updated</u>.

21.   (Currently Amended) A machine-readable medium that provides instructions, which when executed by a processor cause the processor to perform operations comprising:
       establishing communication with a plurality of non-Java-based server nodes;
       establishing communication with a plurality of Java-based server nodes;

transferring packets between the non-Java-based server nodes and the Java-based server nodes; and

ensuring the packets are received by a destination server node <u>by resending the packets if a confirmation of receipt has not been received from a destination server node</u>.

22.   (Currently Amended) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

assigning a service identification associated with each type of services executed on the server nodes; and

maintaining a list of the assigned service identification and corresponding service names<u>, wherein maintaining the service identifications and the service names assists in providing communications between the non-Java-based server nodes and the Java-based server nodes</u>.

23.   (Currently Amended) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

maintaining a list of services <u>currently being</u> executed on the non-Java-based server nodes in a first repository; and

maintaining a list of services <u>currently being</u> executed on the Java-based server nodes in a second repository<u>, wherein each list facilitates distribution of load by indicating currently executing services</u>.

24.   (Currently Amended) The machine-readable medium of claim 23, wherein the operations performed by the processor further comprise:

sending notification of a status of each of the services listed in the first repository to the non-Java-based server nodes<u> such that the maintained lists of services can be updated</u>; and

sending notification of a status of each of the services listed in the second repository to the Java-based server nodes<u> such that the maintained lists of services can be updated</u>.

25.   (Currently Amended) A system comprising:

means for generating a packet <u>on a first non-Java based server node</u> such that a header of the packet specifies an address of a ~~destination~~ <u>first</u> Java-based server node;

means for indicating that the packet is generated by a non-Java based server node;

means for forwarding the packet to intermediate communication means from ~~the one of~~ the first non-Java-based server nodes; and

means for forwarding the packet to the ~~destination~~ first Java-based server node from the intermediate communication means based on the destination address provided in the header of the packet;

means for maintaining a list of services, the list of services includes processes and tasks performed by the first Java-based server node, wherein maintaining the list of service identifications and the service names assists in providing communications between the first non-Java-based server node and the first Java-based server node; and

means for sending notification of a status of each of the listed services to the first Java-based server node, the notification indicates whether the service is running or stopped such that the maintained list of services can be updated.

26.   (Currently Amended) The system of claim 25, further comprising:

means for generating a second packet on a second Java based server node such that a header of the second packet specifies an address of a ~~destination~~ second non-Java-based server node and that the second packet originated from ~~one of~~ a Java-based server nodes;

means for forwarding the second packet to the intermediate communication means from ~~the one of~~ the second Java-based server nodes; and

means for forwarding the second packet to the ~~destination~~ second non-Java-based server node from the intermediate communication means based on the destination address provided in the header of the second packet.

27.   (Currently Amended) The system of claim 26, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the Java-based server nodes wherein maintaining the list assists in providing communications between the Java-based server nodes and the non-Java-based server nodes; and

means for sending notification of a status of each of the listed services to the Java-based server nodes such that the maintained list of services can be updated.

28.   (Currently Amended) The system of claim 27, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the non-Java-based server nodes wherein maintaining the list of service identifications and the service names assists in providing communications between the non-Java-based server nodes and the Java-based server nodes; and

means for sending notification of a status of each of the listed services to the non-Java-based server nodes such that the maintained list of services can be updated.

29.   (Original) The system of claim 25, wherein the intermediate communication means further comprises:

means for establishing communication with a plurality of non-Java-based server nodes;

means for establishing communication with a plurality of Java-based server nodes; and

means for transferring packets between the non-Java-based server nodes and the Java-based server nodes.

30.   (Currently Amended) The system of claim 25, wherein the intermediate communication means further comprises:

means for assigning a service identification associated with each type of services executed on the Java based server nodes and the non-Java based server nodes; and

means for maintaining a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in providing communications between the Java-based server nodes and the non-Java-based server nodes.

31.   (New) A system comprising:

a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from a first non-Java-based server node;

a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from a first Java-based server node;

a message server coupled between the Java-based server nodes and the non-Java-based server nodes to establish communication between the first Java-based server node and the first non-Java-based server node by formatting each packet header such that it is compatible with a format of a destination server, the message server comprising:

a first repository to maintain a list of processes and tasks performed by the plurality of non-Java based server nodes, the first repository includes a plurality of rows each associated with a service performed by a non-Java based server and a plurality of columns which identify attributes associated with each service, the attributes include a server identification to indicate a server performing the service, a service mask to identify the type of service, and a status to indicate the status of the service;

a second repository to maintain a list of processes and tasks performed by the plurality of Java based server nodes, the second repository includes a plurality of rows each associated with a service performed by a Java based server and a plurality of columns which identify attributes associated with each service, the attributes include a server identification to indicate a server performing the service, a service mask to identify the type of service, and a status to indicate the status of the service; and

a third repository to maintain a list of assigned service identifications and their corresponding service names, the third repository includes a plurality of rows each associated with a service and a plurality of columns which identify attributes associated with each service, the attributes include a service name to indicate the name of the service and an assigned service mask to indicate the type of service.